"Express Mail" mailing label number EL544504487US
Date of Deposit September 29, 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner for Patents, Washington, D.C. 20221

Dean E. McConnell, Atty. Reg. No. 44,916

PATENT
Case No. 10022/040

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
PATENT APPLICATION

09656080 - 092900

INVENTOR(S):	Scott R. Sargent
TITLE:	ENVIRONMENT SERVICE ARCHITECTURES FOR NETCENTRIC COMPUTING SYSTEMS
ATTORNEYS:	Dean E. McConnell BRINKS HOFER GILSON & LIONE One Indiana Square, Suite 2425 Indianapolis, Indiana 46204 (317) 636-0886

ENVIRONMENT SERVICE ARCHITECTURES FOR NETCENTRIC COMPUTING SYSTEMS

This application claims the benefit under 35 U.S.C. §119(e) of U.S. provisional application Serial No: 60/156,962 filed on October 1, 1999.

Field of the Invention

The present invention relates generally to business computing systems, and more particularly to environment service architectures for netcentric computing systems.

Background of the Invention

Computer based business solutions have existed for various different types of transactions since the mid-to-late 1960s. During this time period, the technology focused on the use of batch technology. In batch processing, the business user would present a file of transactions to the application. The computer system would then run through the transactions, processing each one, essentially without user intervention. The system would provide reporting at some point in the batch processing. Typically, the reports would be batch printed, which in turn, would be used by the business user to correct the input transactions that were resubmitted along with the next batch of transactions.

In the 1970s, businesses began a transition to on-line, interactive transactions. At a conceptual level, this processing opened up the file of transactions found in batch transactions and allowed the user to submit them one at a time, receiving either immediate confirmation of the success of the transaction or else feedback on the nature of the transaction error. The conceptually simple change of having the user interact with the computer on a transaction-at-a-time basis caused huge changes in the nature of business computing. More important, users saw huge changes in what they could do on a day-to-day basis. Customers were no longer forced to wait for a batch run to process the particular application. In essence, the computer had an impact on the entire work flow of the business user.

Along with the advent of on-line interactive systems, it was equally significant that the systems provided a means for the business user to communicate with others in the business as the day-to-day business went along. This capability was provided on the backbone of a wide area network (WAN). The WAN was in itself a demanding technology during this time period and because of these demands telecommunications groups emerged within organizations, charged with the responsibility to maintain, evolve, and manage the network over a period of time.

The theme of the 1980s was database management systems (DBMSs). Organizations used and applied database technology in the 1970s, but in the 1980s they grew more confident in the application of DBMS technology. Because of the advances in network technology, the focus was now on the sharing of data across organizational and application boundaries. Curiously, database technology did not change the fundamental way in which business processing was done. DBMS made it more convenient to access the data and to ensure that it could be updated while maintaining the integrity of the data.

In the 1990s, technology began to shift toward client/server computing. Client/server computing is a style of computing involving multiple processors, one of which is typically a workstation, and across which a single business transaction is completed. Using the workstation, the transaction entered by the user could now be processed on a keystroke-by-keystroke basis.

Furthermore, there was a change in the communications. With client/server, users could communicate with others in the work group via a local area network (LAN). The LAN permitted workstation-to-workstation communications at speeds of 100 to 1,000 times what was typically available on a WAN. The LAN was a technology that could be grown and evolved in a local office with little need for direct interaction from the telecommunications group.

During the late 1990s, the Internet began to receive widespread use by consumers and businesses. In the business world, the Internet has caused the concept of business users to expand greatly because of the way in which computers are now capable of being interconnected. In addition, the cost of computers has dropped to the point that it is affordable for almost every household to own a computer if they so desire. As such, a need to expand the reach of computing both within and outside the enterprise, and that enables the sharing of data and content between individuals and applications has developed.

To that end, a great need exists amongst existing and newly created business enterprises for computing solutions that capture and take advantage of the advantages and capabilities provided by the Internet.

Summary of the Invention

The present invention discloses an environment services architecture and a method of providing an environment services architecture that is part of a netcentric execution architecture used in a netcentric computing system. The environment services architecture includes runtime services for converting non-compiled computer languages into machine

code during the execution of an application on the netcentric computing system. System services perform system level functions on the netcentric computing system that may be selected from the group consisting of system security services, profile management services, task and memory management services and environment verification services. Application services perform common functions in the netcentric computing system. The common functions may be selected from the group consisting of application security services, error handling/logging services, state management services, code table services, active help services, application integration services and common services.

A component framework service provides components of the netcentric computing system with a standard infrastructure for allowing an application running on components to communicate within and across applications in the netcentric computing system. Operating system services are provided for providing the netcentric computing system with underlying basic computing services. As set forth above, the preferred environment service architecture is designed to provide the netcentric computing system with miscellaneous application and system level services that do not deal directly with managing the user interface, communicating to other programs, or accessing data.

The runtime services include language interpreter services and virtual machine services. The language interpreter services decompose a scripting language into machine code at runtime. The virtual machine services include at least one virtual machine. In the preferred embodiment, the virtual machine services provide a layer of abstraction between applications and the underlying operating system and are used to support operating system independence.

As previously set forth, the system services include system security services, profile management services, environment verification services and task and memory management services. The system security services provide applications with the ability to interact with native security mechanisms that are used by an operating system on the netcentric computing system. The profile management services are used to access and update a plurality of user or application profiles. The environment verification services monitor, identify and validate application integrity before applications are executed on the netcentric computing system. The task and memory management services allow applications or events to control individual computing tasks or processes and manage memory resources in the netcentric computing system.

As set forth above, the application services include application security services, error handling/logging services, state management services, codes table services, active help

services, file services, application integration services and other common services. The application security services may be selected from the group consisting of user access services, data access services and function access services. The error handling/logging services present users of the netcentric computing system with an explanation of errors and logs error events in a database. The state management services enable information to be shared between windows, web pages and applications in said netcentric computing system.

The codes table services enable applications or a client to use externally stored parameters and validation rules on the netcentric computing system. The active help services enable applications to provide assistance to a user or a client for a specific task in said netcentric computing system. The file services enable applications to use, manage and write to files that are located in the netcentric computing system. The application integration interface services provide a gateway for passing context and control of information to an external application. The common services provide a plurality of reusable routines that may be used across a set of applications in said netcentric computing systems.

Further objects and advantages of the present invention will be apparent from the following description, reference being made to the accompanying drawings wherein preferred embodiments of the present invention are clearly shown.

Brief Description of the Drawings

Figure 1 illustrates a netcentric execution architecture for a netcentric computing system that includes a environment services architecture.

Figure 2 illustrates a representative illustration of a netcentric computing system.

Figure 3 illustrates an environment services architecture for a netcentric computing system.

Detailed Description of the Presently Preferred Embodiments of the Invention

Referring to Figs. 1 and 2, the present invention discloses an environment services architecture 10 that is preferentially used in a netcentric execution architecture 11 of a netcentric computing system 12. The physical picture of an illustrative netcentric computing system 12 is illustrated in Fig. 2. In this example, a business enterprise 18 includes at least one client 14, at least one database server 22, at least one firewall 24, at least one application server 26, at least one web server 28 and a local area network (LAN) connection 30, which are electrically connected as illustrated in Fig. 2 to form a network.

As known in the art, LAN connections 30 are comprised of software and various other devices that are used to interconnect various components or computing devices that are located at a first enterprise location 32 within the much larger business enterprise 18. The term LAN connection, as used herein, should be broadly construed to hardware and software that allows clients 14 and servers 22, 26, 28 to be connected together to share data and files as well as the resources available on the netcentric computing system 12. Those skilled in the art would recognize that several components, such as network cards and cabling, may be required to create the LAN connection 30. In addition, those skilled in the art would recognize that various types of LAN connections 30 exist and may be used in the present invention.

For the purpose of the present invention, the firewall 24 is used to isolate internal systems from unwanted intruders. As known in the art, firewalls 24 isolate web servers 28 from all Internet traffic that is not relevant to the netcentric computing system 12. In the preferred embodiment, the only requests allowed through the firewall 24 are for services located on the web servers 28. All requests for other applications (e.g., FTP, Telnet) and other IP addresses are typically blocked by the firewall 24 during operation of the netcentric computing system 12.

The web servers 28 are the primary interface to the clients 14 for all interactions with the applications or services that are provided by the netcentric computing system 12. The main task of the web servers 28 is to authenticate the clients 14, establish a secure connection from the clients 14 to the web servers 28 using encrypted messages, and allow applications the clients 14 are using to transparently access the resources of the netcentric computing system 12. The web servers 28 are responsible for accepting incoming HTTP (Hyper-Text Transfer Protocol) messages and fulfilling the requests. For dynamic HTML (Hyper-Text Markup Language) page generation, requests are forwarded to the application servers 26. During operation, static pages, such as help pages, are preferably generated entirely by the web servers 28. The term client should be construed herein to include remote clients and local clients, unless otherwise specified, as set forth in detail below.

In the preferred embodiment, the primary function of the application servers 26 is to provide a link through which the web servers 28 can interact with the clients 14, trigger business transactions, and send back resulting data to the clients 14. A fundamental role of the application servers 26 is to manage the logical flow of transactions and keep track of the state of sessions. The application servers 26 are also responsible for managing all sessions within the netcentric computing system 12. A session is a period of time in which a client 14

is interacting with, and using, a resource of the netcentric computing system 12, preferentially by accessing the resources through a web browser that is connected with the web server 28.

In the preferred embodiment of the present invention, the main purpose of the database servers 22 is to handle an application log. All requests sent to the web servers 28 and application servers 26, as well as their respective responses, are logged in the application log. The application log is preferentially used for traceability. In the preferred embodiment, requests are logged in the application log directly by the application server 26. Those skilled in the art would recognize that any number of data items can be monitored and kept track of in the application log.

As further illustrated in Fig. 2, a second business enterprise location 34 may be connected with the first business enterprise location 32 using an intranet connection 36. Those skilled in the art would recognize that various intranet connections 36 exist and may be used in the present invention. The intranet connection 36 allows the computing resources of the second business enterprise location 34 to be shared or connected with the computing resources available at the first business enterprise location 32. The term intranet connection, as used herein, should be broadly construed to include software and hardware that is used to connect one server at business location with another server at another business location. Although not illustrated, several other enterprise locations, each containing its own computing resources, may be connected with the netcentric computing system 12.

In the preferred embodiment illustrated in Fig. 2, the firewall 24 of the first business enterprise location 32 is connected with an Internet connection 38 to a plurality of remote clients 39. Preferentially, the remote clients 39 that are connected to the Internet connection 38 access data and resources at the business enterprise 18 by using the Internet connection 38 and a web browser located on the remote client 39. The Internet connection 38 gives the remote clients 39 the ability to gain access to applications, information and data that may be located on the database server 22, the application server 26 and the web server 28. The difference between a "remote" client 39 and a traditional client 14 is that the remote client 39 accesses data and resources of the netcentric computing system 12 only through an Internet connection 38 and not a LAN connection 30 or a intranet connection 36. This allows users on remote clients 39 to access the resources of the netcentric computing system 12 from remote geographic locations anytime, anywhere (such as, at home, a friends house, laptops in hotels, kiosks, library terminals, etc.).

Those skilled in the art would recognize that several methods and devices may be used to provide an Internet connection 38 to the netcentric computing system 12. In addition,

those skilled in the art would recognize that various devices (e.g. – modems, cable modems, etc.) and sub-services (such as Internet Service Providers) may be used to provide remote clients 39 with the Internet connection 38. As used herein, the term Internet connection 38 should be broadly construed to include the hardware and software that enables remote clients 39 and servers 22, 26, 28 to connect with, and communicate on the global information network commonly known as the Internet.

~~Referring to Fig. 1, for a detailed discussion of the other elements of the preferred~~ netcentric execution architecture 11, as well as netcentric computing systems 12, refer to co-pending U.S. patent application Serial Number _____ entitled ARCHITECTURES FOR NETCENTRIC COMPUTING SYSTEMS, which was filed on September 29, 2000 and ~~is hereby incorporated by reference, in its entirety.~~

Referring to Figs. 1 and 3, the present invention discloses an environment services architecture 10 for a netcentric execution architecture 11 that is used in a netcentric computing system 12. As illustrated in Fig. 3, the preferred environment services architecture 10 includes runtime services 40, system services 42, application services 44, component framework services 46 and operating system services 48. As set forth below, the environment services architecture 10 provides services to the netcentric computing system 12 that allows the netcentric computing system 12 to operate optimally to provide business solutions to various enterprises desiring an Internet based solution. The environment services architecture 10 provides miscellaneous application and system level services to the netcentric computing system 12 that do not deal directly with managing the user interface on the clients 14, communicating to other applications, or accessing data in the netcentric computing system 12.

In the preferred embodiment of the present invention, the runtime services 40 are used to convert non-compiled computer languages into machine code during the execution of an application within the netcentric computing system 12. Two sub-services comprise the preferred runtime services 40, which are language interpreter services 50 and virtual machine services 52. The language interpreter services 50 decompose a fourth-generation and/or a scripting language into machine code (executable code) at runtime. The virtual machine services 52 provide virtual machines (VM) to the netcentric computing system 12, which as known in the art are implemented in software on top of an operating system and are used to run applications. The virtual machine services 52 provide a layer of abstraction between applications and the underlying operating system on the client 14 or server 22, 26, 28 and are often used to support operating system independence.

The system services 42 are applications that are used to perform system-level functions in the netcentric computing system. As illustrated in Fig. 3, in the preferred embodiment the system services 42 preferentially include system security services 54, profile management services 56, task and memory management services 58 and environment verification services 60. The system security services 54 allow applications to interact with an operating system's native security mechanisms. The operating system's native security mechanisms include basic services that include the ability to log in, log off, authenticate to the operating system and enforce access control to system resources and executables. As known in the art, several basic services, as defined herein, are provided by commonly used operating systems manufacturers that can provide various levels of security to clients 14, 39 and servers 22, 26, 28.

The profile management services 56 of the systems services 42 are used to access and update local or remote clients 14, 39 user or application profiles by the netcentric computing system 12. User profiles, for example, can be used to store a variety of information, from the user's language and color preferences to basic job function information. This information may be used and shared with other services in the netcentric computing system 12 as well. Those skilled in the art would recognize that several types of information may be stored and located in user profiles.

The environment verification services 58 ensure functionality by monitoring, identifying and validating environment integrity prior to and during the execution of an application (e.g., free disk space, monitor resolution and correct version) in the netcentric computing system 12. The environment verification services 58 are invoked when an application begins processing or when a component is called. Applications can use the environment verification services 58 to verify that the correct versions of the required netcentric execution architecture 11 components and other application components are available. In some client 14, 39/server 22, 26, 28 applications, it may be necessary to implement the environment verification services 58 to ensure that the client 14, 39 and server 22, 26, 28 applications are of a compatible release level before continuing.

ActiveX frameworks can be included in the netcentric computing system 12 for providing services for automatic installation and upgrade of ActiveX controls. Internet Explorer, Microsoft's web browser, is integrated with Windows OS, so ActiveX controls can be automatically installed and automatically upgraded on the clients 14, 39 without the developer adding any additional code. This service may be used in the netcentric computing

system 12 to upgrade and ensure that applications being provided by the servers 22, 26, 28 can run properly on the clients 14, 39.

In the preferred embodiment, the task and memory management services 60 are provided for allowing applications and/or other events in the netcentric computing system 12 to control individual computer tasks or processes and manage memory. The task and memory management services 60 provide applications for scheduling, starting, stopping and restarting both client 14, 39 and server 22, 26, 28 tasks (e.g., software agents). Memory management, the allocating and freeing of system resources, is one of the more error-prone development activities when using 3GL development tools. Using task and memory management services 60 for memory-handling functions reduces the frequency of errors, which are difficult to debug. Those skilled in the art would recognize that task and memory handling applications will vary from system to system, depending on the needs and requirements of each particular netcentric computing system 12.

The application services 44 include applications that perform common functions within the netcentric computing system 12. The application services 44 use applications that provide functions that can apply to one application or across multiple applications. As illustrated in Fig. 3, in the preferred embodiment of the present invention, the application services 44 preferentially include application security services 62, error handling/logging services 64, state management services 66, code table services 68, active help services 70, file services 72, application integration interface services 74 and other common services 76. Each of the above-referenced services is discussed in detail below.

As set forth in Fig. 3, the preferred application services 44 include application security services 62. In addition to the system security services 54, such as logging into a network (typically provided in LAN connection cases), the environment services architecture 10 there are also application security services 62 associated with specific applications on the clients 14, 39 and servers 22, 26, 28 that include user access services, data access services and function access services. User access services provide a set of common functions that limits application access to specific users within a company or external customers. Data access services provide a set of common functions that limits access to specific data within an application to specific users or user types (e.g., secretary, manager). Function access services provide a set of common functions that limits access to specific functions within an application to specific users or user types (e.g., secretary, manager).

In netcentric computing systems 10 application security becomes more critical because there are more types of users (e.g., employees and customers) and additional types of

transactions (e.g., e-Commerce and help desks). In traditional client/server environments, most users of the companies computing systems are employees of the company. In netcentric computing systems 10, there are also external users (e.g., vendors and registered users) and the general public, which may access the netcentric computing system 12 from a variety of remote clients 39 located in vast geographic locations. Different types of users can be given different application security requirements limiting what data they can see and what functions they can execute in the netcentric computing system. Also, new types of transactions, such as verifying credit when doing e-Commerce transactions, also require additional application security services.

The error handling/logging services 64 support the handling of fatal and nonfatal hardware and software errors for an application in the netcentric computing system 12. The error handling/logging services 64 present users on clients 14, 39 with an understandable explanation of what has happened and coordinating with other services to ensure that transactions and data are restored to a consistent state. In addition, the error handling/logging services 64 support the logging of informational, error and warning messages. Logging services record application and user activities in enough detail to satisfy any audit trail requirements or to assist the systems support team in recreating the sequence of events that led to an error.

Primarily, there are three types of errors: system, architecture and application. System errors occur when the application is being executed and some kind of serious system-level incompatibility is encountered, as a result of which the application cannot proceed with its normal execution. These errors can result from such things as memory/resource depletion, database access problems, network problems or printer-related problems.

Architecture errors are those that occur during the normal execution of the application. They are generated in architecture functions that are built by a project architecture team to isolate the developers from complex coding, to streamline the development effort by reusing common services. These architecture functions perform services such as database calls and state management.

Application errors are also those that occur during the normal execution of the application. They are generally related to business logic errors such as invalid date, invalid price and so forth. Typically, an application is written using a combination of various programming languages (e.g., Visual Basic and C). Therefore, a common error-handling routine should be written in a language that can be called from any other language used in the application.

Logging must also be done, however, to mitigate problems, centralize logs and create a standard, usable log format. Third-party logs should be mapped into the central format before any analysis is attempted. In netcentric computing systems 10, errors are rarely logged on the client 14, 39 (although an exception may be for an intranet-type application) and are virtually always logged on the servers 22, 26, 28. Logging can add a great deal of stress to a web server 28 and log files can quickly grow very large. Consequently, system administrators should not plan to log all errors, but, instead, only those deemed necessary for processing exceptions.

The state management services 66 enable information to be passed or shared among windows and web pages and/or across applications in the netcentric computing system 12. For example, suppose that several fields in an application need to be passed from one window to another window on the client's 14, 39 desktop. In pseudo-conversation mainframe 3270-style applications, passing data from one screen to another screen was done using context management services, which provided the ability to store information on a host computer (here, the term "context management" refers to storing state information on the server, not the client). Client/server architectures simplified or eliminated the need for context management (storing state information on the server) and created a need to store state information on the client. In the netcentric computing system 12, this type of state management (i.e., data sharing) is preferentially done on the client 14, 39 using hidden fields, global variables, messages, files or local databases.

The popularity of the Internet's HTTP protocol has revived the need for implementing some form of context management services (storing state information on the web server 28). The HTTP protocol is a stateless protocol. Every connection is negotiated from scratch, not just at the page level, but for every element on the page. In the preferred netcentric computing system 12, the web server 28 does not maintain a session connection with the client 14, 39 nor save any information between exchanges of information (i.e., web page submits or requests) with the clients 14, 39. Each HTTP exchange is a completely independent event. Therefore, information entered into one HTML form must be saved by the associated server application somewhere where it can be accessed by subsequent programs in a conversation. The preferred state management services 66 implement state management on both the client 14, 39 and servers 22, 26, 28 in the environment services architecture 10.

The codes table services 68 enable applications on clients 14, 39 to utilize externally stored parameters and validation rules in the netcentric computing system 12. For example,

an application may be designed to retrieve the tax rate for the state of Illinois. When the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the "State Tax Table" and then retrieves the tax rate for Illinois from the web server 28. Note that codes tables provide an additional degree of flexibility. If the tax rates change, the data simply needs to be updated on the web server 28; no application logic needs to be modified.

In situations where an application requires extensive use of codes tables, the codes table services 68 provides the application developers with a set of APIs that can be used to create code/decode tables. This component also provides the option of caching all or parts of the codes table in the application memory on the client 14, 39 for easier and faster access in the future. Code/decode information can be stored at any layer of an n-tier architecture– client 14, 39 databases or server 22, 26, 28 databases. The decision needs to be based upon codes table size and number, information update frequency and write access to the client 14, 39.

The active help services 70 enable an application to provide assistance to a user for a specific task or set of tasks. Context-sensitive help is preferentially used in applications of the netcentric computing system 12; however, this can imply more active support than just hitting the F1 key for example. Preferentially, each active help service application that is located in the active help services 70 is designed to include help that is aware of the user's environment, process and context; in this sense, it can be called "active." The preferred active help services 70 may include components such as wizards (which walk users through a new process), stored or real-time multimedia support and on-demand computer-based training.

The file services 72 enable applications to use, manage, and write to files that may be located on a variety of platforms in a variety of locations within the netcentric computing system 12. File services 72 give clients 14, 39 remote access to files as if they were located locally, thereby providing transparent access to files for the client 14, 39. Those skilled in the art would recognize that the file services 72 will vary from enterprise to enterprise, depending on the needs and requirements of the particular netcentric computer system 12.

The application integration interface services 74 provide a method or gateway for passing context and control of information to an external application in the netcentric computing system 12. The application integration interface services 74 specify how information will be passed, which is typically from the web server 28 to the client 14, 39, and defines the interface by which other applications can expect to receive information. External

applications in this context could include anything from integrated performance support systems to ERP systems such as SAP or Peoplesoft, to external custom applications that have been previously developed. Where possible, application integration interfaces services 74 should make use of the component model defined by the project to broker information (i.e., OLE/COM interfaces) as opposed to custom building data-sharing modules. Those skilled in the art would recognize that various application integration interface services 74 may be readily incorporated and used in the preferred environment services architecture 10.

The common services 76 component of the application services 44 is a catchall category for a plurality of reusable routines that are useful across a set of applications that may be used in the netcentric computing system 12. Some examples of common services include date routines, time zone conversions and field validation routines. The examples set forth above have been set forth for illustrative purposes only and should not be construed as a limitation of the present invention. Those skilled in the art would recognize that several common services 76 exist and may be used in the application services 44 of the environment services architecture 10.

The component framework services 46 provide a standard infrastructure for building components so that they can communicate within an application and across applications, on the same machine (i.e.- client 14, 39 or server 22, 26, 28) or on multiple machines across a network, to work together. COM/DCOM and CORBA are the two leading component industry standards. These standards define how components should be built and how they should communicate and interact with each other.

Object Request Broker (ORB) services, based on COM/DCOM and CORBA, focus on how components communicate. The preferred component framework services 42, which are preferentially also based on CORBA and COM/DCOM, focus on how components should be built. Currently, two of the most dominant component frameworks include active X/OLE and JavaBeans.

Active X and Object Linking and Embedding (OLE) are implementations of COM/DCOM. ActiveX is a collection of facilities forming a framework for components to work together and interact. ActiveX divides the world into two kinds of components: controls and containers. Controls are relatively independent components that present well-defined interfaces or methods that containers and other components can call. Containers implement the part of the ActiveX protocol that allows for them to host and interact with components, forming a kind of back plane into which controls can be plugged. ActiveX is a scaled-down version of OLE for the Internet.

OLE provides a framework to build applications from component modules and defines the way in which applications interact using data transfer, drag-and-drop and scripting. OLE is a set of common services that allows components to collaborate intelligently. In creating ActiveX from OLE2.0, Microsoft enhanced the framework to address some of the special needs of web-style computing. Microsoft's web browser, Internet Explorer, is an ActiveX container. Therefore, any ActiveX control can be downloaded to, and plugged into, the browser. This allows for executable components to be interleaved with HTML content and downloaded as needed by the web browser.

JavaBeans is Sun Microsystems' proposed framework for building Java components and containers. The intent is to develop and API standard that will allow components developed in Java (or beans), to be embedded in completing container frameworks including ActiveX or OpenDoc. JavaBeans API makes it easier to create reusable components in the Java language.

Other component frameworks include OpenDoc and One. CI Labs, formed in 1993, created the OpenDoc architecture to provide a cross-platform alternative component framework, independent of Microsoft's OLE. The OpenDoc architecture is constructed from various technologies supplied by its founding members: IBM, Apple and Word Perfect. The technologies include Bento (Apple's object storage model), Open Scripting Architecture (OSA-Apple's scripting architecture) and SOM/DSOM (IBM's System Object Model/Distributed SOM). IBM's SOM architecture provides analogous services to that of Microsoft's DCOM architecture.

OpenDoc provides an open compound document infrastructure based on CORBA. It uses CORBA as its object model for inter-component communications. OpenDoc architecture provides services analogous to those provided by OLE, and OpenDoc components can also interoperate with OLE components. The OpenDoc equivalent of an object is termed a "part." Each type of part has its own editor, and the OpenDoc architecture has responsibility for handling the communications between the distinct parts.

Open Network Environment (ONE) is an object-oriented software framework from Netscape Communications for use with Internet clients and servers. It enables the integrating of web clients and servers with other enterprise resources and data. By supporting CORBA, ONE-enabled systems will be able to link with object software from a wide array of vendors, including IBM, Sun Microsystems, Digital Equipment and Hewlett-Packard. Netscape is positioning ONE as an alternative to Microsoft's Distributed Common Object Model (DCOM). ONE also complies with Sun Microsystems' Java technology.

The operating system services 48 are the underlying services – such as multi-tasking, paging and memory allocation – that are typically provided by today's modern operating systems. As known in the art, the operating system services 48 are typically used by clients 14, 39 to perform basic computing tasks such as providing a desktop and web browser applications that allow clients 14, 39 to connect to the web servers 28. The operating system services 48 provide the clients 14, 39 and servers 22, 26, 28 of the netcentric computing system 12 with underlying basic computing services. Where necessary, an additional layer of APIs may be provided to gain either operating system independence or a higher level of abstraction for application programmers. Those skilled in the art would recognize that various operating system services 48 exist and may be used in the preferred environment services architecture 10 to provide basic underlying computing capabilities to the netcentric computing system 12.

While the invention has been described in its currently best known modes of operation and embodiments, other modes and embodiments of the invention will be apparent to those skilled in the art and are contemplated. For other features, advantages and combinations of the present invention refer to U.S. Provisional Application Serial No: 60/156,962, entitled NETCENTRIC AND CLIENT/SERVER COMPUTING, which is herein incorporated by reference, in its entirety.